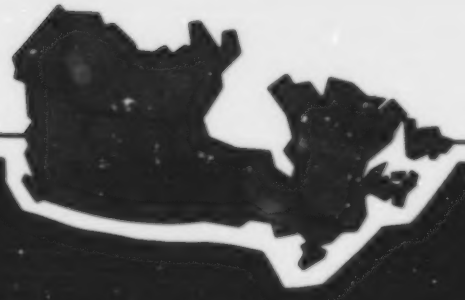
Environment Canada
Canadian Forestry Service

Environnement Canada
Service canadien des forêts

An Electronic "Notebook" for Forestry Application

T. Nieman, W. Kean, and W. Cheliak

Information Report PI-X-38
Petawawa National Forestry Institute



85-4952/

PETAWAWA NATIONAL FORESTRY INSTITUTE

The Petawawa National Forestry Institute (PNFI) was formed on April 1, 1979, as the result of an amalgamation of the Petawawa Forest Experiment Station with the Ottawa-based Forest Management and Forest Fire Research Institutes. The Forestry Statistics and Systems Branch was established at PNFI in 1980.

In common with the rest of the Canadian Forestry Service, the Petawawa National Forestry Institute has as its objective the promotion of better management and wiser use of Canada's forest resource to the economic and social benefit of all Canadians. Because it is a national institute, particular emphasis is placed on problems that transcend regional boundaries or that require special expertise and expensive equipment that cannot be duplicated in CFS regional establishments. Such research is often performed in close cooperation with staff of the regional centres or provincial forest services.

Research at the Institute is in two main areas:

FIRE RESEARCH AND REMOTE SENSING. Every year in Canada large areas of productive forest are destroyed by fire. Research concentrates on studies of forest fire behaviour, the development of new methods of fire control, the evaluation of fire-fighting equipment and retardants, and the development of computerized fire management systems that are rapidly finding applications with fire-fighting agencies across the country. The environmental and economic impact of forest fires and the use of fire as a silvicultural tool for intensive forest management are also studied.

In remote sensing, investigations are made into the application of modern satellite and airborne remote sensing systems to forestry problems. In this respect, the ARIES digital image analysis sytem is proving invaluable.

INTENSIVE FOREST MANAGEMENT. As Canada moves into more intensive management of its forest to meet expected increases in demand for this vital resource, the role of this program will become increasingly important. An extensive reforestation program will require a steady supply of high-quality seed of the desired species. Improved growing stock, obtained through tree breeding and forest genetics research, is highly desirable. Increased emphasis is being placed on using the entire above-ground portion of the tree (biomass), but the effect on the environment of this and other forms of intensive management has to be carefully monitored. Biotechnological methods of improving yield while maintaining site productivity are being investigated.

In support of its research programs, the Institute has at its disposal a 98 km^2 area of forest in the western part of the Petawawa military reserve. Records of experiments and sample plots have been maintained since the 1920s. The forest also serves as a field laboratory for students from local schools, and a visitor centre is operated during the summer months.

The operations of PNFI also includes THE FORESTRY STATISTICS AND SYSTEMS BRANCH (FSSB) which is responsible for the acquisition and publication of national information on the forests of Canada. Through the Canadian Forest Inventory Committee, which is comprised of provincial and federal forestry officials, the FSSB works in close cooperation with provincial forest agencies to improve and standardize the information available on Canada's forest resources.

Through the FORSTATS program, which involves all regional establishments of the Canadian Forestry Service, the FSSB coordinates the acquisition and publication within the CFS of national statistics on the forests of Canada.

Every five years, the FSSB publishes Canada's Forest Inventory; the official report on the location, extent, species, and condition of the forest resource. In addition, the FSSB is working closely with the provinces to expand the information available on changes to the forest from fire, harvesting, insects, and disease and from forest management activities. This information is essential to the development of sound policies for the improved management of this important and renewable natural resource.

AN ELECTRONIC "NOTEBOOK" FOR FORESTRY APPLICATION

Information Report PI-X-38

T. Nieman, W. Kean, and W. Cheliak

Petawawa National Forestry Institute
Canadian Forestry Service
Department of the Environment
1984

©Minister of Supply and Services Canada 1984
Catalogue No. Fo46-11/38-1984E
ISSN 0706-1854
ISBN 0-662-13364-1

Additional copies of this publication can be obtained from

Technical Information and Distribution Centre
Petawawa National Forestry Institute
Environment Canada
Chalk River, Ontario
K0J 1J0

Telephone (613) 589-2880

Cette publication est aussi disponible en français sous le titre **Un "cahier" électronique pour application forestière.**

Contents

i	Abstract/Résumé
1	Introduction
1	Specific hardware functions
1	Datamyte
1	Portable microcomputer
2	Resident minicomputer
2	Characteristics and capability of the Datamyte
2	Physical characteristics
3	Operating characteristics
3	Input modes
4	Description of software library
4	D5(Download)
5	D5B(Download)
5	D4 (Upload)
5	D6 (Identifier Generator)
5	Data collection
5	Local data collection
5	Remote data collection
9	Experience and results
11	Appendix I
13	The computer programs
41	Appendix II
41	Communication between host, portable computers, and Datamyte

Abstract

This report describes an electronic data collection system that facilitates the collection of measurement data from genetic research plantations established in eastern Canada. It has been tested during the past year, locally and at remote field locations, and has replaced the pencil and paper method of data collection. Data are collected rapidly and direct transfer to computer files permits immediate analysis and results from measurement data.The system offers significant savings in time, effort, and cost over manual methods. It has potential for use in many forestry-related fields.

Résumé

Ce rapport décrit un système de collecte de données électronique qui facilite la cueillette des données de dendométrie provenant des centres de recherche génétique établis dans l'est du Canada. Ce système a été mis à l'essai l'année dernière, sur place et dans des centres éloignés; il remplace maintenant la méthode classique (crayon et papier) d'effectuer la collecte des données. Les données sont recueillies rapidement et leur transfert direct aux fichiers informatiques permet d'obtenir des analyses et des résultats immédiats. Le système offre une appréciable économie de temps, d'effort et d'argent si on le compare aux méthodes manuelles. Il peut trouver aussi des applications dans de nombreux secteurs connexes à la foresterie.

AN ELECTRONIC "NOTEBOOK" FOR FORESTRY APPLICATION

INTRODUCTION

Long term forest genetics research often involves collection of large amounts of data from experimental plantations and/or natural forest stands. Until recently, these data have been collected by recording observations on a hard copy coding form with subsequent keypunching to computer cards or a computer file. This operation is very time consuming and costly. As well, keypunching can and does introduce errors into data sets from mis-keyed sequences as well as ambiguities in handwriting. To facilitate recording and transfer of the large amounts of data collected annually by members of the Tree Genetics and Breeding project at the Petawawa National Forestry Institute (PNFI), an electronic data collection system was developed and tested for a variety of diverse applications.

The system involved the use of an electronic "notebook" (Datamyte), a resident minicomputer (PDP 11/44), and a portable microcomputer (Osborne IA). We assessed the system to determine if it would facilitate more efficient data collection and rapid and reliable transfer to the resident computer for storage and processing.

SPECIFIC HARDWARE FUNCTIONS

Datamyte
The Datamyte is a portable electronic data collection and storage device. This device interfaces with the PDP 11/44 minicomputer and the Osborne microcomputer. It is user programmable for a wide variety of applications.

Portable microcomputer
The portable microcomputer functions as an intermediary between the Datamyte and the resident computer. Files of data can be stored on a floppy disk until required at a remote field location. These data can then be downloaded from the floppy disk of the microcomputer to the Datamyte memory. After the current data are collected, they can be moved from the Datamyte memory back to a floppy disk until it is convenient to move them to the resident minicomputer.

T. Nieman is a senior technician, W. Kean is a computer programmer analyst, and W. Cheliak is a research scientist at the Petawawa National Forestry Institute.

Manuscript approved for publication: April 30, 1984

We do not endorse any of the hardware products mentioned in this report. Competitive brands and different models of data recorders, mini and microcomputers are available but were not tested.

Resident minicomputer
The resident minicomputer, a PDP 11/44, is used for analysis of data and as a permanent storage facility for sets of unprocessed data. It can be interfaced directly with the Datamyte as well as with the microcomputer.

CHARACTERISTICS AND CAPABILITY OF THE DATAMYTE

Physical characteristics
Several Datamyte models are available differing primarily in memory storage. The model tested was the Datamyte 1003 (Figure 1) which is a solid state storage device with 64K memory. This unit has a 16 digit liquid crystal display (LCD) with a 23 key alphanumeric keyboard, 12 character keys, and 11 function keys. The Datamyte weighs 1.5 kg and can be hand-held or used with a shoulder strap. Power for the unit is supplied by a rechargeable nickel cadmium battery as well as a built-in standby battery. The standby battery is used to maintain the memory when the main battery is discharged or removed. The keyboard is protected by a transparent membrane which allows data to be collected in virtually any weather. The unit can be operated at temperatures ranging from -40°C to +40°C. At extremely cold temperatures, battery performance is decreased resulting in slower performance of the Datamyte.

The Datamyte interfaces with computer systems and/or terminals via a cable and a RS232-C port or a 20mA current loop interface. Data can be transferred to or from the unit at nominal baud rates ranging from 110 to 4800. Appearance of the transmitted data can be modified by the operator.

Data entered into memory can be displayed for review or editing. In addition, specific records can be located by a built-in search function.

Figure 1. The Datamyte 1003.

Operating characteristics

The Datamyte is operator-programmable, allowing the user flexibility to arrange the unit for virtually any data entry format. Programming of the Datamyte is by four control functions, each of which controls a different aspect of data input or output. The parameters in each of the four control functions are briefly described.

Control 1. Contains all primary functions of the Datamyte including memory status, record section, in-mode and interface device selector, and record number.

Control 2. Controls the transmission options of the Datamyte from or to a terminal, including parity, baud rate, pad characters, and a time delay.

Control 3. Controls the transmission options of the Datamyte to a host computer, including start of text, character count, checksum, end of text, data acknowledgment, and end of transmission.

Control 4. Controls the format for data transmission.

Input modes

Four input modes, which control the type of data to be entered, are accessed through control 1. A brief description of each of these input modes follows.

In-mode 1. (Data + edit) Alpha or numeric data are entered directly into memory. The Datamyte records the data exactly as it was keyed.

In-mode 2. (Data + edit + prompt) Prior to recording data, a series of prompts referred to as the prompt loop must have been entered. The prompts recorded in the loop are displayed in sequence to ensure that data are collected in the correct order. Prompts can be entered to memory directly from the Datamyte keyboard or from a computer terminal.

In-mode 3. (Data + edit + download) and In-mode 4. (Data + edit + prompt + download). With the appropriate software, selected sections of data collected previously and stored on a computer file (called the download list) can be transferred to Datamyte memory. This download list may be used as a check against current records (e.g. presence/absence, change in magnitude) and may be helpful in correcting previous errors. The difference between in-mode 3 and in-mode 4 is that the latter option allows entry of a prompt-loop as well as the download list.

For much of our data collection we use in-mode 4. The download list includes a plot identification (composed of family identity, replication, and tree number) as well as the most recent measurement.

Additionally, the Datamyte can store up to eight separate files and the in-modes can vary for each file. Data can be edited in all four of the

in-modes. Header, prompts, download, and data can be edited only before proceeding to a new section of file, as determined by control function 1.

DESCRIPTION OF SOFTWARE LIBRARY

A software library was developed to permit information to be transferred between any two units (i.e. Datamyte, Osborne, PDP 11/44). Two of these programs, D4(Upload) and D5(Download), exchange information between the Datamyte and the resident computer. Information transfer between the Osborne and the Datamyte is accomplished by a program called D5B(Download) and Micro-Link. Micro-Link (Osborne software) is also used to transmit data from the PDP 11/44 to the Osborne and vice versa. An additional program D6(Identifier-Generator) can be used to generate sequential lists of numbers which can then be transferred to the Datamyte via program D5. Program documentation and listing for each program is supplied in Appendix I. Below is a brief overview of each program.

D5(Download)
This program is used to transfer data or information from a file in the resident minicomputer to Datamyte memory. Usually these data consist of the plot identities and most recent measurements. The program eliminates the need to key this information manually and thus reduces the possibility of errors. As well, the program saves time and provides a predetermined guide to ensure that data are recorded in the proper sequence. The user has the option of specifying in-mode 3 or 4. The settings of the four control functions are displayed and allow entry of header information, prompts (depending upon in-mode), download lists, and format for transmission.

If in-mode 4 is selected, all desired prompts for data are entered, plus one additional prompt which will contain the downloaded list. This "list-prompt" must occur as the first position of the prompt loop. For example, if two variables are to be recorded, three prompts must be entered; the first prompt is for the download list, followed by prompts for the two variables. However, a maximum of 16 characters in total (this includes the prompt name) are reserved for any prompt. After the download list has been transferred successfully to the Datamyte, current data can be recorded by following the prompt sequence.

When using in-mode 3, a download list can be transferred to the Datamyte. However, a prompt-loop sequence cannot be entered under this operating mode. Therefore, provision must be made within the context of the program to allow current data to be entered using this mode. This is accomplished by the use of "variables" rather than prompts. The difference between in-modes 3 and 4 is that by using in-mode 3, one record for each variable is downloaded to the Datamyte memory, whereas when using in-mode 4 a "list prompt" downloads only one record per prompt loop. Again, the maximum number of characters, including reserved space for the current data, cannot exceed the 16 character display.

After the download list has been successfully transmitted to the Datamyte, the program produces a report indicating the number of header,

prompt, and download-data records. The report is displayed on the computer terminal. The Datamyte is now ready to record the current set of data.

D5B(Download)
This program is written in M-BASIC and is used to download information to the Datamyte which have been stored on a floppy disk on the Osborne. This system and program are intended primarily for application at remote locations, as discussed later.

D4(Upload)
After the current set of data are recorded in the Datamyte memory, they can be transmitted to a file on the resident computer for analysis. This transmission is accomplished by use of program D4. The function of the upload program is to indicate appropriate control function settings which will safely transfer data to a file. The appearance of the transmitted data is determined by the operator via control function 4, not by the program.

D6(Identifier-generator)
In some instances it may be possible to use a sequential list of numbers which could identify the unit measured. Program D6 is designed to generate a download-list of identifiers which can then be moved from a file to the Datamyte memory via program D5.

DATA COLLECTION

Local data collection
When data are collected locally (Figure 2), the Datamyte can be interfaced with the resident computer system (Figure 3) to allow lists to be downloaded or data uploaded. These transfers are normally performed on a daily basis using programs D5 and D4.

Remote data collection
Frequently, data must be collected away from the local computing facilities. In our specific situation this can involve trips of up to three weeks during which many different assessments are made. Obviously, it is not possible to store all of the download lists required and new data to be collected on the Datamyte during the entire trip. Initially, a portable terminal (with a built-in acoustic coupler) was used to transmit data to the resident computer over long-distance phone. This procedure was unsatisfactory for a number of reasons. First, the baud rate (speed of transmission) was limited to a slow 300 baud. Secondly, the long-distance calls proved to be very expensive. Finally, because of "noise" in these telephone lines, data often had to be transmitted more than once to ensure that a complete set of data was on file.

To alleviate these problems, we acquired a portable microcomputer (Osborne) to be used with the Datamyte (Figure 4) at remote field sites. Prior to a field trip, download lists are prepared and moved from the resident computer to a floppy disk on the Osborne via a communications link (Micro-Link).

Figure 2. Collecting data with the Datamyte 1003.

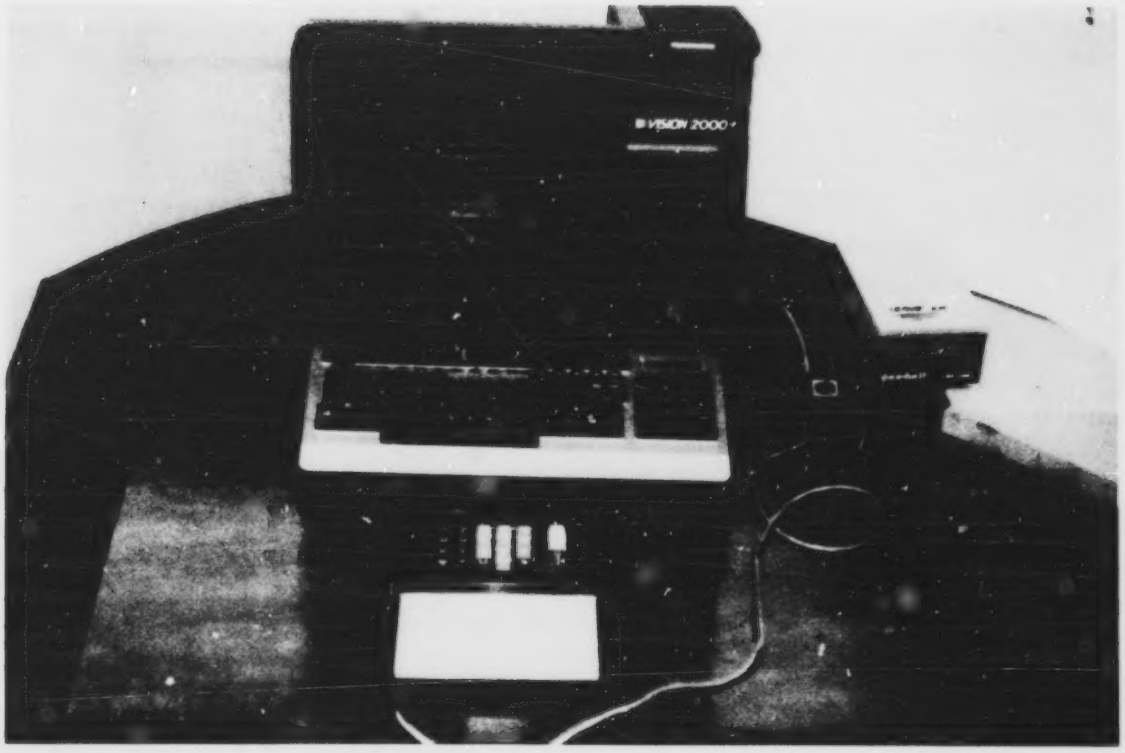


Figure 3. Interfacing the Datamyte 1003 with the resident computer.

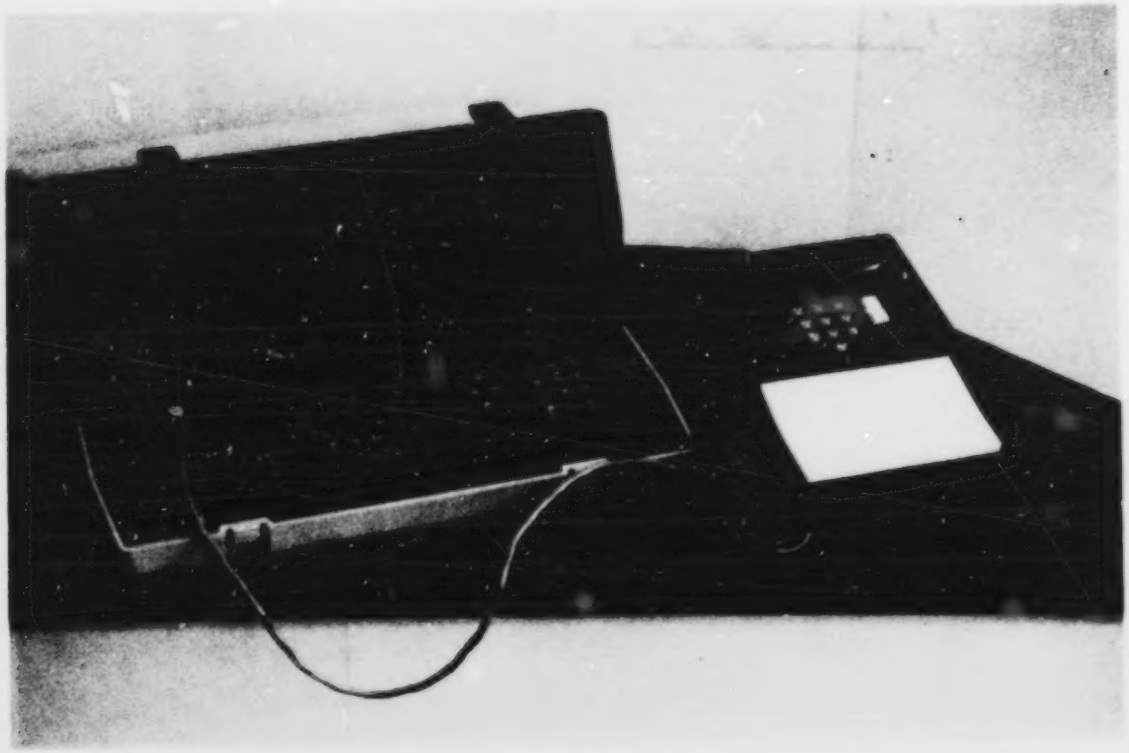


Figure 4. The Osborne portable microcomputer.

When needed, the download list as well as the header and prompt information are moved from diskette to the Datamyte memory via program D5B. After the current data have been collected, they are transferred and stored on a floppy disk via Micro-Link. The data may be summarized and/or analyzed immediately with the microcomputer if required. When local computing facilities are available, the data are moved from the floppy disk to a permanent file on the resident computer via Micro-Link.

Detailed instructions to enable communication between the resident computer, portable Osborne computer, and the Datamyte are described in Appendix II.

EXPERIENCE AND RESULTS

Electronic data collection offers numerous advantages over conventional methods of gathering data. The system saves effort and time by eliminating the time-consuming and expensive keypunching operation. As well, preparation and storage of field tally sheets are not necessary. Some specific advantages of the Datamyte, as used in our application, are briefly discussed.

The Datamyte is completely portable, lightweight, and weatherproof. Data can therefore be collected in virtually any type of application and/or field season. We have found that electronic entry of data, after a brief training period, is faster and more legible than recording data on tally sheets. The possibility of collecting data in an incorrect sequence, or omitting data, is minimized by the use of the programmable prompt sequence as well as the download list.

For our field plantation assessments, the Datamyte with 64K memory is capable of storing information for approximately 3200 trees, alloting 20 digits per tree including the download list. On an average, it takes 2-3 normal working days to fill the Datamyte memory. It is however, good practice to transfer data collected to a resident or microcomputer on a daily basis.

After the data are transmitted to the resident minicomputer, they can be partially proofed using appropriate software. These programs can check for anomalies in the data. Alpha sequences for numeric values (or vice versa) or values out-of-range (high and low limits) are two examples. These proofing functions depend upon specific applications.

Although no "hard copy" is produced with this unit during data collection, the memory is protected. Internal memory is maintained by a back-up power supply whenever the main power source is discharged or accidentally removed. This can keep the contents of the memory intact for a period of up to 15 hours. The possibility of accidental memory loss, while not impossible, is reduced by a elaborate sequence of keys that have to be pressed in a specific order. Thus, clearing the memory is normally a deliberate action.

The Datamyte can also be used as a stand-alone terminal for direct communication with the host computer. However, because of the sixteen character display limit, we do not recommend the unit for this purpose.

To date, all models produced by the Electro-General company can be updated to accommodate the most recent developments. Servicing, if necessary, is available from factory authorized dealers.

We have found that the limited display is the major disadvantage of this particular unit. When using downloaded lists, we have occasionally had to limit the amount of information to be stored.

In our application, records are stored and must continue to be accessible for decades. Obviously, this necessitates a well-structured and organized data storage and retrieval system. This is true for both "hard" and "soft" copies of data.

Appropriate software and hardware must be readily available at all times for effective use of an electronic data collection system. As well, the system must be an integral part of program operation and be utilized on a regular basis. Otherwise, much time can be spent re-learning procedures and regaining efficiency.

Although this electronic data collection system was developed for use in research plantations, we feel that it has potential for widespread use in many forestry related fields including timber cruising, scaling, regeneration surveys, and routine laboratory activities.

APPENDIX I

The Computer Programs

Three of the computer programs presented here, D4(Upload), D5(Download), and D6(Identifier generator) were originally developed at Agriculture Canada.

In 1982 these programs were modified, documented, and made operational on the Petawawa National Forestry Institute's DEC PDP11/44 with the RSX-11M Operating System. They are written in Fortran-77/RSX.

Program D5 uses one extension to the ANSI Standard Fortran, the ENCODE statement, which is found in the PDP-11 FORTRAN-77/RSX. The function of ENCODE is described at its first occasion in the program. This function is normally available in most Fortrans but possibly under another statement name.

Programs D4 and D5 run effectively when built not-checkpointable.

Program D5B(Download) was written at PNFI for use with the portable Osborne microcomputer.

Listings for each of the programs, including documentation for D5, follow. Copies of the programs can be obtained by sending a request to:

Petawawa National Forestry Institute
Attn: General Computing Services
Chalk River, Ontario
K0J 1J0

The software will be provided on an "as is" basis. The Petawawa National Forestry Institute and its employees assume no responsibility for the accuracy or reliability of the programs.

Program D4(Upload):	Uploads Datamyte files to the host computer.
Data Input:	The input data is received from the Datamyte and uploaded to the host computer. The program uploads headers, prompts, and data.
Data Output:	The program prompts for the output file name and uploads one Datamyte file to that name. It then prompts for another output file name.
Constraints and limitations:	The program is limited to 132 characters per record. The settings for the 4 Datamyte control functions are displayed.
Program D4.FTN:	See next page

```
C     PROGRAM D4.FTN
CC
C     THIS PROGRAM UPLOADS FILES FROM THE DATAMYTE TO THE COMPUTER.
C
C     THE PROGRAM WAS RECEIVED FROM AGRICULTURE CANADA AND
C     HAS BEEN DOCUMENTED, MODIFIED, CHECKED AND MADE OPERATIONAL
C     ON A PDP 11/44  BY WILLIAM KEAN  JULY 1982.
CC
C               The Government of Canada
C               Department of the Environment
C               Canadian Forestry Service
C
C                  Notice of Copyright
C                  -------------------
C
C       This software, in its entirety, is the property of the
C       Government of Canada. The following software is not
C       to be used, reproduced, copied in whole or in part
C       nor is the information contained in the software to
C       be disclosed without the written permission of the
C
C               Petawawa National Forestry Institute (PNFI)
C               Canadian Forestry Service
C               Environment Canada
C               Chalk River, Ontario
C               K0J 1J0
C
C                  Credit Request and Disclaimer
C                  -----------------------------
C
C       If and when permission for use is granted, suitable
C       credit for the use of the following software should be
C       included in any resulting publication. Also PNFI or its
C       employees assume no responsibility for the accuracy or
C       reliability of the following software.
C
C     USE THE FOLLOWING STEPS TO BUILD D4 ON A RSX-11M OPERATING SYSTEM.
C     F77 D4/NOTR,D4/SP=D4
C     TKB>D4/-CP/-AL,D4=D4
CC
CC
C     THE PROGRAM PROMPTS FOR THE OUTPUT FILENAME
C     DATAMYTE CONTROLS SHOULD BE SET AS FOLLOWS:
C         CONTROL 1     15 22 XX
C         CONTROL 2     3510000
C         CONTROL 3     00006  0
C         CONTROL 4     02002102
CC
      BYTE RECIN(132),FILNAM(20),Y,Y1
      DATA Y/'Y'/
      WRITE(5,9)
 9    FORMAT(/1X'THIS PROGRAM D4 IS USED TO UPLOAD DATAMYTE FILES TO
     + THE COMPUTER.'/)
      WRITE(5,8)
 8    FORMAT(/1X'DATAMYTE CONTROLS SHOULD BE SET AS FOLLOWS:'/
     +1X'CONTROL 1    15 22 XX'/
     +1X'CONTROL 2    3510000'/
     +1X'CONTROL 3    00006  0'/
     +1X'CONTROL 4    02002102'//)
      WRITE(5,14)
 14   FORMAT(/1X'CTRL 2-2 MUST BE SET AS  FOLLOWS FOR DIFFERENT BAUD
     + RATES'
```

```
     +/1X'3 FOR  300 BAUD'/1X'5 FOR 1200 BAUD'/1X'6 FOR 2400 BAUD'
     +/1X'7 FOR 4800 BAUD'//)
1     WRITE(5,10)
   10 FORMAT(//1H$,'ENTER NAME OF OUTPUT FILE:')
      READ(5,20)FILNAM
   20 FORMAT(20A1)
C
C     "0 = NULL CHARACTER.
C
      FILNAM(20)="0
      OPEN(UNIT=1,NAME=FILNAM,TYPE='NEW',CARRIAGECONTROL='LIST')
      WRITE(5,45)
   45 FORMAT(1H0,'TURN THE DATAMYTE ON AND SELECT THE FILE ON THE'/
     +1X'DATAMYTE THAT YOU WANT BY SETTING CTRL 1-7.'/
     +1H0'SET DATAMYTE CONTROLS AND '/
     -         1X,'KEY CLEAR, CONTROL, XMIT')
   40 READ(5,50)ILEN,(RECIN(I),I=1,ILEN)
   50 FORMAT(Q,132A1)
      IF(ILEN.LT.2)GO TO 99
      WRITE(1,50)ILEN,(RECIN(I),I=1,ILEN)
C
C     THE 0 IS SENT TO TELL THE DATAMYTE THAT THE COMPUTER IS READY TO
C     RECEIVE MORE DATA.
C     C3-5-6 SETS THE ACKNOWLEDGMENT TO ANY NUMBERIC CHARACTER.
C
      WRITE(5,55)
   55 FORMAT(2H 0)
      GO TO 40
   99 CLOSE(UNIT=1)
      WRITE(5,11)
11    FORMAT(/1X'TURN DATAMYTE OFF AND THEN REPLY TO THE FOLLOWING:'
     +/1X'DO YOU WANT TO UPLOAD ANOTHER FILE? [Y/N]: '$)
      READ(5,12) Y1
12    FORMAT(A1)
      IF(Y1.EQ.Y) GO TO 1
30    STOP
      END
```

Program D5(Download):	This program extracts specific fields from disk files or generates identifiers, and preloads the Datamyte or writes the data to a disk file for transfer to a personal computer.
Data Input:	The program asks for headers and prompts to be entered via the keyboard. These are downloaded to the Datamyte. The data input file can be: (1) The character file created by the EDITOR, a computer program or a file uploaded by program D4 from the Datamyte. (2) A binary file containing sequential identifiers created by program D6.
Data Output:	The download lists can be output to the Datamyte in two formats depending on whether input mode 3 or 4 is used.
Constraints and limitations:	The record length cannot exceed 16 characters including separators added by the program. There can be a maximum of 5 identifiers including test number if used.
Program D5.FTN:	See next page

```
C     PROGRAM D5.FTN
C
C     THIS PROGRAM DOWNLOADS FILES TO THE DATAMYTE.
C
C     THE PROGRAM WAS OBTAINED FROM AGRICULTURE CANADA
C     AND HAS BEEN EXTENSIVELY DOCUMENTED, MODIFIED, CHECKED AND MADE
C     OPERATIONAL ON A PDP 11/44 BY W.A. KEAN IN OCTOBER 1982.
C
C               The Government of Canada
C               Department of the Environment
C               Canadian Forestry Service
C
C                  Notice of Copyright
C                  -------------------
C
C       This software, in its entirety, is the property of the
C       Government of Canada. The following software is not
C       to be used, reproduced, copied in whole or in part
C       nor is the information contained in the software to
C       be disclosed without the written permission of the
C
C               Petawawa National Forestry Institute (PNFI)
C               Canadian Forestry Service
C               Environment Canada
C               Chalk River, Ontario
C               K0J 1J0
C
C                  Credit Request and Disclaimer
C                  -----------------------------
C
C       If and when permission for use is granted, suitable
C       credit for the use of the following software should be
C       included in any resulting publication. Also PNFI or its
C       employees assume no responsibility for the accuracy or
C       reliability of the following software.
C
C     USE THE FOLLOWING STEPS TO BUILD D5 ON A RSX-11M OPERATING SYSTEM.
C     F77 D5/NOTR,D5/SP=D5
C     TKB>D5/-CP/-AL,D5=D5
C
CC
CC
C     VARIABLES DICTIONARY
C
C     ANSA = Y OR N TO DATAMYTE READY FOR DOWNLOAD.
C     ANSH = Y OR N TO ENTER HEADERS.
C     ANSP = Y OR N TO ENTER PROMPTS.
C     ANSR = Y OR N TO RELOAD A FILE TRANSMITTED FROM THE DATAMYTE.
C     ANSTYP = 1 IF FILE CREATED BY D6 (IDENTIFIER GENERATOR) OR
C              2 IF CHARACTER FORMAT FILE CREATE BY AN EDITOR OR OTHER MEANS.
C     ANS1 = Y OR N TO DOES A TEST NO. EXIST WITH EACH RECORD IN THE FILE.
C     ANS2 = Y OR N TO DO YOU WISH A TEST NUMBER TO BE ADDED TO OUTPUT RECORD.
C     ANS3 = Y OR N TO DO YOU WISH TO SELECT RECORDS BY TEST NUMBER IN THE FILE.
C     ANS4 = Y OR N TO DO YOU WISH TO DOWNLOAD ANOTHER FILE.
C     ANS5 = Y OR N TO DO YOU WISH TO CREATE A FILE IN YOU ACCOUNT ON THE MAIN
C            COMPUTER TO BE LOADED TO YOUR DISK ON YOUR PERSONAL COMPUTER
C            AT A LATER DATE USING YOUR PERSONAL COMPUTER'S SOFTWARE.
C     AST = ASTERISK.
C     CA = CHARACTER A.
C     C0 = 0
C     FILNAM = INPUT FILE NAME.
C     FILOUT = OUTPUT FILE NAME.
```

```
C     FMT(5) = (I  )
C     FORM = FORMAT FOR THE DATA IE: FOR DATA OF THE FORM -99.99
C            THE DATA FORMAT WOULD BE H00H00
C     HEADER = HEADER SUBROUTINE.
C     HYPH = -
C     IB = CONTAINS THE NUMBER OF DIGITS IN THE IDENTIFIER TO BE
C          DOWNLOADED FROM THE FILE CREATED BY D6 (IDENTIFIER GENERATOR).
C     ICOL(5) = ARRAY VARIABLE OF 5. CONTAINING THE STARTING COLUMN
C               FOR EACH OF UP TO 5 IDENTIFIERS.
C     ICONT1 = NO. OF RECORDS TRANSMITTED.
C     ICONT2 = NO. OF RECORDS SUCCESSFULLY DOWNLOADED.
C     ICONT3 = TOTAL NUMBER OF RECORDS IN ERROR.
C     IDATCT = NUMBER OF SPACES RESERVED FOR THE DATA IN THE DATAMYTE.
C     IDCT = NUMBER OF IDENTIFIERS (INCLUDING TEST NUMBER IF USED)
C            TO BE DOWNLOADED FROM THE FILE CREATED BY D6 (IDENTIFIER
C            GENERATOR), MAX OF 5.
C     IDLEN = NUMBER OF CHARACTERS IN VARIABLE FORM.
C     IENO = END RECORD NUMBER OF OUTPUT RECORD.
C     IETX = ETX = "3   END OF TEXT.
C     IHDCT = NUMBER OF HEADER RECORDS DOWNLOADED.
C     ILEN = NUMBER OF CHARACTERS IN ARRAY RECIN.
C     INC = INCREMENT OF OUTPUT RECORD.
C     INREC(132) = RECORD READ FROM THE CHARACTER FILE ON THE COMPUTER.
C     INV = CONTAINS ACTUAL RECORDS SENT UNSUCCESSFULLY TO A MAX OF 20.
C     IPID(5) = CONTAINS THE IDENTIIFIERS.
C     IPRCT = NO. OF PROMPTS DOWNLOADED.
C     IR = NUMBER OF CHARACTERS IN ARRAY RECIN.
C     IRCT = NUMBER OF RECORDS RETRANSMITTED.
C     ISNO = START RECORD NUMBER OF OUTPUT RECORD.
C     ISR = THE POSITION IN ARRAY RECIN WHICH CONTAINS THE NUMBER
C           OF THE VARIABLE / PLOT DESCRIPTION.
C     ISTX = STX = "2   START OF TEXT.
C     ITCOL(5) = ARRAY VARIABLE OF 5. CONTAINING THE LAST COLUMN FOR
C                EACH OF UP TO 5 IDENTIFIERS.
C     ITEST = TEST NUMBER FOR OUTPUT RECORD WHICH CAN BE READ FROM A
C             COMPUTER FILE OR ENTERED AT THE TERMINAL AND MUST BE IN
C             THE LAST IDENTIFIER POSTION.
C     IVAR = IS THE NUMBER OF VARIABLES PER PLOT DESCRIPTION TO BE
C            GENERATED USING THE SAME PLOT DESCRIPTION. THE ONLY
C            DIFFERENCE BEING THE VARIABLE NUMBER WHICH IS INSERTED IN THE OUTPUT
C            RECORD AND THE VARIABLE NUMBER MUST BE > 0 AND < 10.
C     IVCT = IS THE VARIABLE NUMBER / PLOT DESCRIPTION TO BE INSERTED
C            AND IS INCREMENTED BY ONE UNTIL IT REACHES THE NUMBER OF
C            VARIABLES REQUESTED.
C     LID(5) = LENGTH OF EACH IDENTIFIER FIELD (INCLUDING TEST
C              NUMBER IF USED).
C     LIM = USED TO DETERMINE THE NUMBER OF DIGITS IN THE IDENTIFIER
C           CREATED BY D6 (IDENTIFIER GENERATOR).
C     NCHARS = NUMBER OF CHARACTERS IN THE FILNAM AND FILOUT.
C     PROMPT = PROMPT SUBROUTINE.
C     RECIN = THE RECORD BEING ENTERED TO THE DATAMYTE.
C     REP = REPLY FROM THE DATAMYTE WHICH IN THIS SET UP IS AN A.
C     RS = CONAINS ACTUAL RECORDS RESENT TO A MAXIMUM OF 20.
C     SEND = SEND SUBROUTINE.
C     SP = SPACE.
CC
CC
CC
      LOGICAL A1 RECIN(16),FILNAM(40),FILOUT(40),C0,FORM(14),ISTX,IETX,
     -    FMT(5),SP,INREC(132),AST,HYPH,INV(20,16),RS(20,16)
      DATA FMT/'(','I',' ',' ',')'/,SP/' '/,AST/'A'/,HYPH/'-'/
      DATA C0/'0'/,YES/'Y'/,ISTX/"2/,IETX/"3/
      DIMENSION ICOL(5),ITCOL(5),LID(5)
```

```
      INTEGER ANSTYP
      INTEGER*4 IPID(5),LIM
1     ANSTYP=0
      WRITE(5,96)
96    FORMAT(X'DO YOU WISH TO CREATE A FILE IN YOUR ACCOUNT ON THE MAIN'
     +/X'COMPUTER TO BE LOADED TO THE DISK ON YOUR PERSONAL COMPUTER'/
     +X'AT A LATER DATE USING YOUR PERSONAL COMPUTER''S SOFTWARE.
     + [Y/N]:   '$)
      READ(5,110) ANS5
      IF(ANS5.EQ.YES) THEN
      WRITE(5,97)
   97 FORMAT(/1H$,'ENTER NAME OF OUTPUT FILE:   ')
      READ(5,240)NCHARS,FILOUT
  240 FORMAT(Q,40A1)
      FILOUT(NCHARS+1)="0
      OPEN(UNIT=3,NAME=FILOUT,TYPE='NEW',CARRIAGECONTROL='LIST')
      END IF
CC
C     THIS IF BYPASSES THE CHOICE OF ENTERING HEADERS, PROMPTS AND
C     UPLOADING A FILE TRANSMITTED
C     FROM THE DATAMYTE IF CREATING A FILE ON THE SYSTEM TO
C     BE READ BY A PERSONAL COMPUTERS SOFTWARE.
CC
      IF(ANS5.EQ.YES) GO TO 400
      WRITE(5,100)
  100 FORMAT(/1H$,'DO YOU WISH TO DOWNLOAD HEADER RECORDS [Y/N]:   ')
      READ(5,110)ANSH
  110 FORMAT(A1)
      IF(ANSH.EQ.YES)CALL HEADER(RECIN,RS,INV,IHDCT,IRCT)
      WRITE(5,200)
  200 FORMAT(/X,'DO YOU WISH TO DOWNLOAD PROMPTS [Y/N]:'/
     +1H$,'ANSWER Y IF THE DATA IS TO BE COLLECTED IN INPUT-MODE 4:   ')
      READ(5,110)ANSP
      IF(ANSP.EQ.YES)CALL PROMPT(RECIN,RS,INV,IPRCT,IRCT)
  210 IF(ANS5.EQ.YES) GO TO 400
      WRITE(5,220)
  220 FORMAT(/1H$,'DO YOU WISH TO RELOAD A FILE TRANSMITTED FROM THE ',
     -     'DATAMYTE [Y/N]:   ')
      READ(5,110)ANSR
      IF(ANSR.NE.YES)GO TO 400
CC
C     THIS SECTION RELOADS A FILE TRANSMITTED FROM THE DATAMYTE.
CC
      WRITE(5,230)
  230 FORMAT(/1H$,'ENTER NAME OF FILE TO BE RELOADED:   ')
      READ(5,240)NCHARS,FILNAM
      FILNAM(NCHARS+1)="0
      OPEN(UNIT=2,NAME=FILNAM,TYPE='OLD',READONLY)
  245 WRITE(5,250)
  250 FORMAT(/1X'IN ORDER TO DOWNLOAD THE DATAMYTE WITH A FILE OF'
     +/1X'IDENTIFIERS CONTROLS SHOULD BE SET AS FOR PROMPTS, EXCEPT'
     +/1X'C1-2 SHOULD BE 3 IF INPUT-MODE IS 4 AND SHOULD BE 4 IF'
     +/1X'INPUT-MODE IS 3.'/1X'KEY CLEAR, CONTROL, RECIEVE.'/
     +/1X,'DATAMYTE READY FOR DOWNLOAD - ANSWER [Y/N]:  '$)
      READ(5,110)ANSA
      IF(ANSA.NE.YES)GO TO 245
  255 READ(2,260,END=1000)ILEN,(RECIN(I),I=1,ILEN)
  260 FORMAT(Q,16A1)
      ICONT1=ICONT1+1
      CALL SEND(RECIN,ILEN,ICONT2,RS,INV,IRCT,ICONT3,ANS5)
      GO TO 255
CC
C     DETERMINES WHETHER OR NOT A FILE EXISTS OTHER
```

```
C     THAN THE ONE TRANSMITTED BY THE DATAMYTE.
CC
  400 WRITE(5,410)
  410 FORMAT('1H$,'NAME OF INPUT FILE (IF NO INPUT FILE EXISTS
     + ENTER 0)?   ')
      READ(5,240)NCHARS,FILNAM
      FILNAM(NCHARS+1)='0
      IF(FILNAM(1).NE.C0)GO TO 425
CC
C     IF NO FILE EXISTS THEN THIS PROGRAM WILL GENERATE & PRELOAD
C     SEQUENTIAL NUMBERS. ACCEPTS STARTING AND LAST RECORD NUMBER
C     PLUS INCREMENT.
CC
      WRITE(5,411)
  411 FORMAT(/1H$,10X,'START RECORD NO: ')
      READ(5,412) ISNO
  412 FORMAT(I4)
      WRITE(5,413)
  413 FORMAT(1H$,10X,'END RECORD NO: ')
      READ(5,412)IENO
      WRITE(5,414)
  414 FORMAT(1H$,10X,'INCREMENT: ')
      READ(5,412)INC
      IENO=IENO+INC
      GO TO 525
CC
C     DETERMININES IF THE FILE IS UNFORMATTED AND CREATED BY D6 (IDENTIFIER
C     GENERATOR), OR A FORMATTED FILE.
CC
  425 WRITE(5,428)
  428 FORMAT(1H0,'INPUT FILE (1) CREATED BY D6 (IDENTIFIER GENERATOR)'/
     -        1X,'        OR (2) CHARACTER FORMAT FILE'/
     -       1H$,'ANSWER "1" OR "2" : ')
      READ(5,430)ANSTYP
  430 FORMAT(I1)
      IF(ANSTYP.EQ.1) GO TO 510
CC
C     THIS SECTION DETERMINES THE NUMBER & POSITION IN THE FILE OF THE
C     IDENTIFIERS TO BE DOWNLOADED FROM A FORMATTED FILE.
CC
      OPEN(UNIT=1,NAME=FILNAM,TYPE='OLD',READONLY)
      WRITE(5,440)
  440 FORMAT(/1X,'NO. OF IDENTIFIERS TO BE DOWLOADED FROM FILE,
     +/1X'MUST BE > 0 AND < 6 IN FORMAT I2: '$)
      READ(5,450)IDCT
  450 FORMAT(I2)
      DO 500 I=1,IDCT
      WRITE(5,470)I
  470 FORMAT(1X,'IDENTIFIER FIELD ',I2/
     -      1H$,'          STARTS IN COL.: ')
      READ(5,480)ICOL(I)
  480 FORMAT(I2)
      WRITE(5,490)
  490 FORMAT(1H$,'             ENDS IN COL.: ')
      READ(5,480)ITCOL(I)
  500 CONTINUE
      GO TO 580
CC
C     THIS SECTION OPENS THE FILE CREATED BY D6 (IDENTIFIER GENERATOR) AND
C     DETERMINES IF A TEST NUMBER IS PRESENT IN THE FILE.
CC
  510 OPEN(UNIT=1,NAME=FILNAM,TYPE='OLD',FORM='UNFORMATTED'
     -     RECORDTYPE='FIXED',READONLY)
```

```
      WRITE(5,520)
  520 FORMAT(/1H$,'DOES A TEST NO. EXIST WITH EACH RECORD [Y/N]:   ')
      READ(5,110)ANS1
      IF(ANS1.EQ.YES) GO TO 550
CC
C     ALLOWS YOU TO ENTER A TEST NUMBER INTO THE RECORD CREATED BY D6.
CC
  525 WRITE(5,530)
  530 FORMAT(/1H$,'DO YOU WISH A TEST NO. TO BE ADDED
     -     'TO THE OUTPUT RECORD [Y/N]:   ')
      READ(5,110)ANS2
      IF(ANS2.NE.YES)GO TO 580
      WRITE(5,535)
  535 FORMAT(1H$,'ENTER TEST NUMBER YOU WISH ADDED IN FORMAT I4: ')
      READ(5,540)ITEST
  540 FORMAT(I4)
      GO TO 580
CC
C     THIS SECTION DETERMINES IF YOU WANT TO SELECT RECORDS FROM
C     THE COMPUTER FILE CREATED BY D6 BY TEST NUMBER.
CC
  550 WRITE(5,560)
  560 FORMAT(/1H$,'DO YOU WISH TO SELECT RECORDS IN THE
     + COMPUTER FILE, BY TEST NUMBER [Y/N]:   ')
      READ(5,110)ANS3
      IF(ANS3.NE.YES)GO TO 580
      WRITE(5,570)
  570 FORMAT(1H$,'ENTER TEST NUMBER TO BE DOWNLOAD IN FORMAT I4?')
      READ(5,540)ITEST
  580 WRITE(5,590)
  590 FORMAT(/1X,'ENTER HOW MANY VARIABLES / PLOT DESCRIPTION
     + IN FORMAT I4.'/1X'MUST BE > 0 & < 10     '$)
      READ(5,540)IVAR
      IF((IVAR.GT.0).AND.(IVAR.LT.10))GO TO 600
      WRITE(5,595)
  595 FORMAT(1X,'NUMBER OF VARIABLE MUST BE > 0 AND < 10')
      GO TO 580
  600 WRITE(5,610)
  610 FORMAT(/1X,'ENTER NUMBER OF SPACES TO BE RESERVED FOR THE DATA
     +IN FORMAT I4 ? '/1X'IF INPUT-MODE 4 IS BEING USED ENTER 0   '$)
      READ(5,540)IDATCT
CC
C     IF NO SPACES ARE TO BE LEFT FOR THE DATA GO TO 640.
CC
      IF(IDATCT.EQ.0)GO TO 640
      WRITE(5,620)
  620 FORMAT(/' DATA FORMAT ?'
     +/' IE:  IF DATA TO BE RECORDED IS OF THE FORM -99.99'
     +/'      THEN THE DATA FORMAT WOULD BE H00H00'/)
      READ(5,630)IDLEN,FORM
  630 FORMAT(Q,14A1)
  640 IF(ANSTYP.EQ.2)GO TO 700
CC
C     THIS SECTION USED TO DETERMINE THE LENGTH OF EACH IDENTIFIER FIELD.
CC
      WRITE(5,650)
  650 FORMAT(1HO,'TYPE LENGTH OF EACH IDENTIFIER FIELD'
     -     '(INCLUDING TEST NUMBER)'/
     -     1X,'UP TO 5 INTEGERS IN FORMAT 5I2')
      READ(5,660)LID
  660 FORMAT(5I2)
      IDCT=0
      DO 670 I=1,5
```

```
      IF(LID(I).GT.0)IDCT=IDCT+1
  670 CONTINUE
CC
C     DETERMINES IF THE DATAMYTE IS READY FOR DOWNLOADING.
CC
  700 IF(ANS5.NE.YES) THEN
      WRITE(5,250)
      READ(5,110)ANSA
      IF(ANSA.NE.YES)GO TO 700
      END IF
CC
C     CLEARS ARRAY RECIN.
CC
      DO 7001 I=1,16
CC
C     "0    =    NULL CHARACTER.
CC
 7001 RECIN(I)="0
CC
C     IF FILNAM(1) = 0 THERE IS NO FILE TO BE DOWNLOADED, GO TO 705.
CC
  701 IF(FILNAM(1).EQ.C0)GO TO 705
CC
C     IF FILE CREATED BY D6 (IDENTIFIER GENERATOR) GO TO 710.
CC
      IF(ANSTYP.EQ.1)GO TO 710
CC
C     READS IN ONE RECORD FROM THE CHARACTER FILE TO A MAXIMUM
C     OF 132 CHARACTERS.
CC
      READ(1,680,END=1000)INREC
  680 FORMAT(132A1)
      IR=1
CC
C     LOADS THE IDENTIFIERS IN ARRAY INREC INTO ARRAY RECIN
C     WITH ASTERICK SEPARATORS.
CC
      DO 690 I=1,IDCT
      IF(I.EQ.1)GO TO 682
CC
C     PLACES AN ASTERICK BETWEEN EACH IDENTIFIER.
CC
      RECIN(IR)=AST
      IR=IR+1
CC
C     PLACES INDIVIDUAL IDENTIFIERS IN SEQUENCE IN ARRAY RECIN.
CC
  682 DO 685 J=ICOL(I),ITCOL(I)
      RECIN(IR)=INREC(J)
      IR=IR+1
  685 CONTINUE
  690 CONTINUE
      GO TO 802
CC
C     WHEN THERE IS NO FILE TO DOWNLOAD THIS SECTION OF THE PROGRAM
C     CREATES SEQUENTIAL NUMBERS TO BE DOWNLOADED.
CC
  705 IPID(1)=ISNO
      ISNO=ISNO+INC
      IF(ISNO.GT.IENO)GO TO 1000
      GO TO 720
CC
C     READS FILE CREATED BY D6 (IDENTIFIER GENERATOR).
```

```
CC
  710 READ(1,END=1000)IPID
CC
C     IF YOU ARE NOT SELECTING BY TEST NUMBER ASSOCIATED WITH
C     THE FILE RECORD GO TO 720; OTHERWISE SELECTS BY TEST
C     NUMBER OUT OF THE FILE.
CC
      IF((ANS3.NE.YES).OR.(ANS1.NE.YES))GO TO 720
CC
C     CHECKS TO SEE IF FILE RECORD CORRESPONDS TO SELECTED TEST
C     NUMBER.
CC
      IF(IPID(IDCT).EQ.ITEST)GO TO 720
      GO TO 710
CC
C     THIS IF PLACES THE TEST NUMBER IN THE IPID ARRAY IN THE
C     LAST USED POSITION IN THE ARRAY WHEN THERE IS NO TEST NUMBER
C     READ FROM THE FILE.
CC
  720 IF(ANS2.EQ.YES)IPID(IDCT)=ITEST
CC
C     SETUP RECIN WITH IDENTIFIERS CREATED BY D6 (IDENTIFIER GENERATOR).
CC
      IR=1
      DO 800 I=1,IDCT
      IF(I.EQ.1)GO TO 725
      RECIN(IR)=AST
      IR=IR+1
CC
C     DETERMINES THE NUMBER OF DIGITS IN THE VALUE TO BE DOWNLOADED
C     IB CONTAINS THE NUMBER OF DIGITS IN THE VALUE.
CC
  725 IB=1
      LIM=9
  730 IF(IPID(I).LE.LIM)GO TO 740
      IB=IB+1
      LIM=LIM*10+9
      GO TO 730
CC
C     IF THE VALUE DOES NOT FILL UP THE SIZE ALLOCATED BY THE OPERATOR
C     FOR THE VARIABLE THE PROGRAM LEFT HAND FILLS THE FIELD WITH ZEROS.
CC
  740 IF(IB.EQ.LID(I))GO TO 750
      DO 745 J=IB+1,LID(I)
      RECIN(IR)=C0
      IR=IR+1
  745 CONTINUE
CC
C     THIS SECTION PUTS THE NUMBER OF DIGITS IN THE FORMAT
C     IE: IF IB = 2 PUTS THE 2 IN TO THE I FORMAT AND MAKES IT I2.
C     THIS IS DONE FOR EACH PIECE OF DATA TO FORM THE FORMAT FOR
C     STATEMENT 760.
CC
  750 IF(IB.GT.9)GO TO 755
  752 FORMAT(I1)
      ENCODE(1,752,FMT(3))IB
CC
C     ENCODE.
C     ENCODE(c,f,b)list
C     where:
C     c  is an integer expression. The c is the number of characters
C        (bytes) to be translated to character form.
C     f  is a format identifier.
```

```
C     b  is the name of an array, array element, variable, or character
C        substring reference. In the ENCODE statement, b receives the
C        characters after translation to external form.
C     list is an I/O list. In the ENCODE statement, the I/O list contains
C          the data to be translated to character form.
C     ENCODE is a PDP 11 language element. The ENCODE statement translates
C     the list elements to character form according to the format
C     specifier, and stores the characters in b. If fewer than c characters
C     are transmitted, the remaining character positions are filled with
C     spaces.
CC
      FMT(4)=SP
      GO TO 760
  755 ENCODE(2,757,FMT(3))IB
  757 FORMAT(I2)
CC
C     PUTS THE VALUE IN IPID(I) INTO ARRAY RECIN USING THE
C     APPROPRIATE FORMAT.
CC
  760 ENCODE(IB,FMT,RECIN(IR))IPID(I)
      IR=IR+IB
  800 CONTINUE
CC
C     IF VARIABLES / PLOT DESCRIPTION IS GREATER THAN ONE
C     PUTS A 1 INTO RECIN TO INDICATE VARIABLE 1; OTHERWISE,
C     IT DOESN'T PUT A VARIABLE NUMBER INTO RECIN.
CC
  802 IF(IVAR.EQ.1)GO TO 810
      IVCT=1
CC
C     PUTS AN ASTRICKS IN AS A SEPERATOR IN ARRAY RECIN.
CC
      RECIN(IR)=AST
      IR=IR+1
      ISR=IR
CC
C     PUTS THE 1 IN AFTER THE ASTRICKS IN ARRAY RECIN.
CC
  805 ENCODE(1,752,RECIN(IR))IVCT
      IR=IR+1
  810 IF(IDATCT.EQ.0)GO TO 825
CC
C     PUTS A - IN ARRAY RECIN IF SPACES ARE TO BE RESERVED
C     IN THE DATAMYTE FOR DATA (INPUT-MODE 3).
CC
      RECIN(IR)=HYPH
      IR=IR+1
CC
C     PUTS IN THE FORMAT FOR THE DATA INTO ARRAY RECIN.
CC
      DO 815 J=1,IDATCT
      RECIN(IR)=FORM(J)
      IR=IR+1
  815 CONTINUE
  825 IR=IR-1
CC
C     LOOPS BETWEEN HERE AND THE GO TO 830
C     EQUAL TO THE NUMBER OF VARIABLES REQUESTED / PLOT
C     DESCRIPTION.
CC
  830 ICONT1=ICONT1+1
      CALL SEND(RECIN,IR,ICONT2,RS,INV,IRCT,ICONT3,ANS5)
      IF(IVAR.LT.2)GO TO 701
```

```
      IF(IVCT.EQ.IVAR)GO TO 701
      IVCT=IVCT+1
      ENCODE(1,752,RECIN(ISR))IVCT
      GO TO 830
CC
C     THIS SECTION SUMMARIZES THE DOWNLOADING OF A FILE
C     TO THE DATAMYTE AND PROMPTS TO SEE IF YOU WANT TO
C     LOAD ANOTHER FILE.
CC
 1000 WRITE(5,1010)IHDCT,IPRCT,ICONT1,ICONT2,ICONT3
 1010 FORMAT(1H0,'          ***END OF DOWNLOAD***'/
     -        1X,'      NO. OF HEADER RECORDS DOWNLOADED',I4/
     -        1X,'         NO. OF PROMPTS DOWNLOADED',I4/
     -        1X,'        NO. OF RECORDS TRANSMITTED',I4/
     -        1X,'NO. OF RECORDS SUCCESSFULLY DOWNLOADED',I4/
     -        1X,'              NO. OF RECORDS IN ERROR',I4)
      IF(ICONT3.EQ.0)GO TO 1100
      WRITE(5,1020)
 1020 FORMAT(1H0,'THE FOLLOWING RECORDS WERE NOT DOWNLOADED ',
     -     'SUCCESSFULLY:-'/)
      IF(ICONT3.GT.20)ICONT3=20
      DO 1050 I=1,ICONT3
      WRITE(5,1030)(INV(I,J),J=1,16)
 1030 FORMAT(1X,16A1)
 1050 CONTINUE
 1100 IF(IRCT.EQ.0)GO TO 1200
      WRITE(5,1110)
 1110 FORMAT(1H0,'THE FOLLOWING RECORDS WERE RETRANSMITTED:-')
      DO 1130 I=1,IRCT
      WRITE(5,1120)(RS(I,J),J=1,16)
 1120 FORMAT(1X,16A1)
 1130 CONTINUE
 1200 WRITE(5,1210)
 1210 FORMAT(1H$,'DO YOU WISH TO DOWNLOAD ANOTHER FILE ?')
      READ(5,110)ANS4
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      ICONT1=0
      ICONT2=0
      ICONT3=0
      IRCT=0
      IF(ANS4.EQ.YES)GO TO 1
      IF(ANS5.NE.YES) THEN
CC
C     CHARACTER COUNT OF ZERO WITH THE APPROPRIATE CHECKSUM OF
C     146 IS SENT TO THE DATAMYTE TO SIGNAL THE END OF DOWNLOADING
CC
      WRITE(5,1250)ISTX,IETX
 1250 FORMAT(1X,A1,'000146',A1)
      END IF
      STOP
      END
```

```
      SUBROUTINE HEADER(RECIN,RS,INV,IHDCT,IRCT)
      LOGICAL *1 RECIN(16)
      IHDCT=0
      WRITE(5,100)
  100 FORMAT(1H0,'SET DATAMYTE C1-2 TO 1; KEY CLEAR, CONTROL, RECV'/
     -     1X,'TYPE HEADER RECORDS TO BE DOWNLOADED'/
     -     1X,'- ONE RECORD PER LINE'/
     -     1X,'TYPE RETURN TO TERMINATE HEADERS')
  110 WRITE(5,120)
  120 FORMAT(1H$,'HEADER:')
      READ(5,130)ILEN,(RECIN(I),I=1,ILEN)
  130 FORMAT(Q,16A1)
      IF(ILEN.EQ.0)GO TO 500
      CALL SEND(RECIN,ILEN,IHDCT,RS,INV,IRCT,ICONT3,ANS5)
      GO TO 110
  500 ILEN=1
CC
C     "33  =   ESC CHARACTER.
CC
      RECIN(1)="33
      CALL SEND(RECIN,ILEN,IHDCT,RS,INV,IRCT,ICONT3,ANS5)
      RETURN
      END
```

```
      SUBROUTINE PROMPT(RECIN,RS,INV,IPRCT,IRCT)
      LOGICAL *1 RECIN(16)
      IPRCT=0
      WRITE(5,100)
  100 FORMAT(1H0,'SET DATAMYTE C1-2 TO 2; KEY CLEAR, CONTROL, RECV'/
     -     1X,'TYPE PROMPT RECORDS TO BE DOWNLOADED'/
     -     1X,'- ONE RECORD PER LINE'/
     -     1X,'TYPE RETURN TO TERMINATE PROMPTS')
  110 WRITE(5,120)
  120 FORMAT(1H$,'PROMPT:')
      READ(5,130)ILEN,(RECIN(I),I=1,ILEN)
  130 FORMAT(Q,16A1)
      IF(ILEN.EQ.0)GO TO 500
      CALL SEND(RECIN,ILEN,IPRCT,RS,INV,IRCT,ICONT3,ANS5)
      GO TO 110
  500 ILEN=1
      RECIN(1)="33
      CALL SEND(RECIN,ILEN,IPRCT,RS,INV,IRCT,ICONT3,ANS5)
      RETURN
      END
```

```
      SUBROUTINE SEND(RECIN,ILEN,ICT,RS,INV,IRCT,ICONT3,ANS5)
C     ALEN = THE THREE BYTES IN ALEN CONTAIN THE ASCII
C            REPRESENTATION OF THE BCD ENCODED VALUE OF THE NUMBER OF
C            CHARACTERS IN (RECIN).
C     ICT  = NUMBER OF RECORDS SENT FOR THE CURRENT OPERATION IE:
C            HEADERS, PROMPTS.
C     ILEN = NUMBER OF CHARACTERS IN RECIN.
      LOGICAL *1 RECIN(16),ALEN(3),C0,ISTX,IETX,CHK(3),ESC,RS(20,16),
     -    INV(20,16)
      DATA CA/'A'/,C0/'0'/,ISTX/"2/,IETX/"3/,ESC/"33/,YES/'Y'/
      IRSEND=0
CC
C     ENCODES (ILEN) THE NUMBER OF CHARACTERS IN (RECIN) INTO THREE
C     BCD DIGITS AND REPRESENTS THEM BY THEIR ASCII EQUIVALENT.
CC
      ALEN(1)=C0
CC
C     IF THE NUMBER OF CHARACTERS IN THE RECORD IS GREATER THAN
C     9 GO TO 30.
CC
      IF(ILEN.GT.9)GO TO 30
      ALEN(2)=C0
      ENCODE(1,20,ALEN(3))ILEN
   20 FORMAT(I1)
      GO TO 100
   30 ENCODE(2,40,ALEN(2))ILEN
   40 FORMAT(I2)
  100 ITOT=0
CC
C     TOTALS UP THE BINARY VALUES OF THE DATA TO BE SENT TO THE DATAMYTE
C     TO DETERMINE THE CHECKSUM.
CC
      DO 110 I=1,3
      ITOT=ITOT+ALEN(I)
  110 CONTINUE
      DO 120 I=1,ILEN
      ITOT=ITOT+RECIN(I)
  120 CONTINUE
CC
C     THE 2 IS ADDED TO THE CHECKSUM TO ADD IN THE STX
C     CHARACTER WHO'S BINARY VALUE IS 2.
CC
      ITOT=ITOT+2
CC
C     THIS ALGORITHM MAKES ITOT EQUAL TO THE LAST 8 BITS OF ANY
C     INTEGER WORD SIZE IE: *4 OR *2.
CC
      ITOT=ITOT-256*(ITOT/256)
      IF(ITOT.GT.99)GO TO 150
      IF(ITOT.GT.9)GO TO 140
CC
C     ENCODES THE CHECKSUM (ITOT) INTO THREE BCD DIGITS AND REPRESENTS
C     THEM BY THEIR ASCII EQUIVALENT.
CC
      CHK(1)=C0
      CHK(2)=C0
      ENCODE(1,20,CHK(3))ITOT
      GO TO 170
  140 CHK(1)=C0
      ENCODE(2,40,CHK(2))ITOT
      GO TO 170
  150 ENCODE(3,160,CHK(1))ITOT
  160 FORMAT(I3)
```

```
  170 IF(ANS5.NE.YES) THEN
      WRITE(5,180)ISTX,ALEN,(RECIN(I),I=1,ILEN),CHK,IETX
  180 FORMAT(1X,24A1)
      ELSE
      WRITE(3,181)(RECIN(I),I=1,ILEN)
  181 FORMAT(16A1)
      END IF
CC
C     IF RECIN(1) IS AND ESC CHARACTER IT IS ONLY TRANSMITTED
C     ONCE AND THE DATAMYTE SHOULD BEEP AND CLEAR IT'S DISPLAY
C     THIS INDICATES THAT THE HEADER OR PROMPT OPERATION IS
C     COMPLETE AND THE PRGRAM GOES ON TO THE NEXT STEP.
CC
      IF(RECIN(1).EQ.ESC)GO TO 500
CC
C     RECEIVES "A" FOR ACK FROM DATAMYTE WHEN C3-5 IS SET TO 2.
CC
      IF(ANS5.NE.YES) THEN
      READ(5,220)REP
  220 FORMAT(A1)
CC
C     IF THE PROGRAM DOES NOT RECEIVE AN "A" GOES TO 300.
CC
      IF(REP.NE.CA)GO TO 300
      END IF
      ICT=ICT+1
      IF(IRSEND.EQ.0)GO TO 500
      IRCT=IRCT+1
CC
C     STORES THE RECORDS THAT HAD TO BE RETRANSMITTED TO BE
C     ACCEPTED TO A MAXIMUM OF 20.
CC
      DO 240 I=1,16
      RS(IRCT,I)=RECIN(I)
  240 CONTINUE
      GO TO 500
CC
C     TRIES TO SEND THE DATA A MAXIMUM OF 5 TIMES IF THE
C     PROGRAM DOES NOT RECEIVE AN "A" FROM THE DATAMYTE.
C
  300 IRSEND=IRSEND+1
      IF(IRSEND.LT.6)GO TO 170
      ICONT3=ICONT3+1
CC
C     ONLY STORES 20 RECORDS THAT WERE NOT SENT SUCCESSFULLY.
CC
      IF(ICONT3.GT.20)GO TO 500
CC
C     STORES THE RECORD THAT WAS NOT SENT SUCCESSFULLY.
CC
      DO 310 I=1,16
      INV(ICONT3,I)=RECIN(I)
  310 CONTINUE
  500 RETURN
      END
```

Program D5.DOC: This provides a detailed documentation for use with program D5(Download).

See next page

DOCUMENTATION FOR PROGRAM: D5 (DOWNLOAD)

PURPOSE:

TO PRELOAD THE DATAMYTE PORTABLE DATA ENTRY TERMINAL (PDET) WITH INFORMATION STORED ON COMPUTER FILES OR GENERATED IDENTIFIERS PRIOR TO THE COLLECTION OF NEW DATA.

INPUT REQUIRED:

ALL QUESTIONS SHOULD BE ANSWERED WITH 'Y' FOR YES AND 'N' FOR NO.

1) "DO YOU WISH TO CREATE A FILE IN YOUR ACCOUNT ON THE MAIN COMPUTER TO BE LOADED TO THE DISK ON YOUR PERSONAL COMPUTER AT A LATER DATE USING YOUR PERSONAL COMPUTER'S SOFTWARE."

2) "DO YOU WISH TO DOWNLOAD HEADER RECORDS"
ANSWER 'Y' IF YOU WISH TO LOAD ONE OR MORE RECORDS INTO THE DATAMYTE TO IDENTIFY THE CONTENTS OF THE FILE. IN THIS CASE THE DATAMYTE CONTROLS SHOULD BE SET TO DOWNLOAD HEADER RECORDS - SEE OPERATION SECTION.
PROGRAM REQUESTS "HEADER?"
REPLY WITH ALPHANUMERIC STRING.
HEADER PROMPT WILL BE REPEATED UNTIL CARRIAGE RETURN IS ENTERED.

3) "DO YOU WISH TO DOWNLOAD PROMPTS"
ANSWER 'Y' IF DATA IS TO BE COLLECTED IN INPUT-MODE 4.
DATAMYTE CONTROLS SHOULD BE SET TO DOWNLOAD PROMPTS - SEE OPERATION SECTION.
PROGRAM REQUESTS "PROMPT?"
REPLY WITH AN ALPHANUMERIC STRING TO BE USED TO LABEL FIELD TO BE ENTERED.
PROMPT WILL BE REPEATED UNTIL CARRIAGE RETURN IS ENTERED.

4.1) "DO YOU WISH TO RELOAD A FILE TRANSMITTED FROM THE DATAMYTE"
ANSWER "Y" IF YOU WISH TO DOWNLOAD THE DATAMYTE WITH A FILE WHICH WAS PREVIOUSLY TRANSMITTED FROM THE PDET VIA PROGRAM D4 (UPLOAD). FIELD WIDTH CANNOT EXCEED 16 CHARACTERS.
4.2) "NAME OF FILE TO BE RELOADED"
THE PROGRAM WILL THEN CONTINUE FROM SECTION 12.

5.1) NAME OF INPUT FILE (IF NO INPUT FILE EXISTS ENTER 0)
THE INPUT FILE CAN BE EITHER A TERMINAL FORMAT FILE CREATED BY AN EDITOR OR A PROGRAM OR AN UNFORMATTED FILE CREATED BY PROGRAM D6 (IDENTIFIER GENERATOR).
5.2) "INPUT FILE 1) CREATED BY D6 (IDENTIFIER GENERATOR)
OR 2) TERMINAL FORMAT FILE
ANSWER '1' OR '2' "
REPLY '1' IF THE INPUT FILE WAS CREATED BY D6 (IDENTIFIER GENERATOR).
PROGRAM THEN CONTINUES FROM SECTION 7.
REPLY '2' IF THE INPUT FILE IS A TERMINAL FORMAT FILE.
THE PROGRAM WILL THEN ASK FOR "NUMBER OF IDENTIFIERS TO BE DOWNLOADED". REPLY WITH AN INTEGER FROM 1 TO 5.
FOR EACH IDENTIFIER THE PROGRAM WILL THEN PROMPT FOR THE FOLLOWING INFORMATION.
"STARTS IN COL."
"ENDS IN COL."
IDENTIFIERS MAY BE ALPHANUMERIC.
PROGRAM WILL THEN CONTINUE FROM SECTION 8.

6) IF NO INPUT FILE EXISTS THEN THIS PROGRAM CAN GENERATE & PRELOAD SEQUENTIAL NUMBERS.

THE PROGRAM WILL REQUEST THE FOLLOWING NUMERIC VALUES:
5.1) START RECORD NO.
5.2) END RECORD NO.
5.3) INCREMENT - VALUE TO BE ADDED TO THE CURRENT RECORD NO.
TO PRODUCE THE NEXT RECORD NO.

7) IF AN INPUT FILE, CREATED BY PROGRAM D6, ALREADY EXISTS THEN THE USER HAS THE OPTION OF ASSOCIATING A TEST NUMBER WITH EACH SET OF DATA. THIS OPTION CAN BE SELECTED WHEN THE USER WISHES TO DISTINGUISH BETWEEN SEVERAL SETS OF DATA TO BE PRELOADED. THE TEST NUMBER CAN EITHER BE ADDED TO THE PRELOAD RECORD BY THE D5 (DOWNLOAD) PROGRAM OR CAN BE THE LAST IDENTIFIER FIELD IN THE INPUT RECORD. THE PROGRAM REQUESTS THE FOLLOWING INFORMATION:
7.1) "DOES A TEST NO. EXIST WITH EACH RECORD Y/N". ANSWER 'Y' IF THE LAST IDENTIFIER FIELD IN THE INPUT RECORD IS A TEST NUMBER, OTHERWISE ANSWER 'N'.
7.2) IF THERE IS NO TEST NUMBER ASSOCIATED WITH EACH RECORD THE PROGRAM REQUESTS:
"DO YOU WISH A TEST NO. TO BE ADDED TO THE O/P RECORD Y/N".
ANSWER 'Y' OR 'N'.
IF ANSWER IS 'Y' THE PROGRAM PROMPTS FOR A NUMERIC TEST NUMBER.
7.3) "DO YOU WISH TO SELECT RECORDS BY TEST NUMBER Y/N". ANSWER 'Y' IF YOU WISH TO SELECT PART OF THE INPUT FILE BY THE TEST NUMBER THE PROGRAM WILL PROMPT FOR THE TEST NUMBER TO BE DOWNLOADED.

8) "HOW MANY VARIABLES / PLOT DESCRIPTION?"
USED ONLY WHEN USING MODE 3 ON THE DATAMYTE.
REPLY WITH THE NUMBER OF VARIABLES TO BE RECORDED FOR EACH PLOT DESCRIPTION.
ONE RECORD WILL BE DOWNLOADED FOR EACH VARIABLE TO BE ENTERED.
IF MORE THAN ONE VARIABLE IS REQUESTED THEN THE VARIABLE NUMBER IS ADDED TO THE RECORD.

E.G.
IF MEASURING HEIGHT AND MATURITY ON A PARTICULAR PLANT WITHIN A PLOT, THEN RECORDS WILL BE DOWNLOADED AS FOLLOWS:

```
01^01^1-0000
01^01^2-0000
01^02^1-0000
01^02^2-0000  ...ETC...
```

THE RIGHTMOST IDENTIFIER IS THE VARIABLE NUMBER, VARIABLE 1 COULD BE HEIGHT AND VARIABLE 2 MATURITY.

9) "HOW MANY SPACES TO BE RESERVED FOR DATA?"
IF INPUT-MODE 4 IS TO BE USED TO COLLECT DATA THEN REPLY '0'
IF USING INPUT-MODE 3 THEN REPLY GIVING THE NUMBER OF POSITIONS REQUIRED IN THE OUTPUT RECORD TO COLLECT THE DATA.

10) "DATA FORMAT"
THE USER MUST SPECIFY THE FORMAT OF THE DATA TO BE RECORDED. EACH CHARACTER MUST BE DEFINED AS NUMERIC (REPRESENTED BY 0) OR ALPHABETIC (REPRESENTED BY H).
E.G.
IF DATA TO BE RECORDED IS OF THE FORM -99.99
THEN THE DATA FORMAT WOULD BE H00H00

11) "TYPE LENGTH OF EACH IDENTIFIER FIELD (INCLUDING TEST NUMBER) - UP TO 5 INTEGERS SEPARATED BY COMMAS."
TYPE UP TO 5 NUMBERS SEPARATED BY COMMAS TO DEFINE THE NUMBER OF POSITIONS REQUIRED IN THE OUTPUT RECORD BY EACH IDENTIFIER IN THE PLOT DESCRIPTION (INCLUDING THE TEST NUMBER IF USED).

THIS QUESTION IS ONLY ASKED IF YOU ARE USING A FILE CREATED
BY PROGRAM D6 OR IF YOU HAVE ANSWERED THE QUESTIONS IN SECTION 6.

12) "DATAMYTE READY FOR DOWNLOAD Y/N"
ANSWER 'Y' OR 'N'
THE DATAMYTE SHOULD BE SET UP AS IN OPERATION SECTION:
AFTER SETTING UP THE CONTROL FIELDS PRESS CLEAR AND RECV.
THE DATAMYTE DISPLAY WILL SHOW
DATA 0000 IF INPUT-MODE 3
LIST 0000 IF INPUT-MODE 4
THE MACHINE IS THEN READY FOR DOWNLOADING.

13) AT THE END OF PRELOADING A PARTICULAR FILE OR TEST THE
PROGRAM WILL ASK:
"DO YOU WISH TO DOWNLOAD ANOTHER FILE Y/N."
ANSWER 'N' TO TERMINATE THE PROGRAM.
ANSWER 'Y' TO CONTINUE.

OUTPUT:

1) RECORDS ARE DOWNLOADED TO THE DATAMYTE IN THE FOLLOWING
FORMAT:
ID1^ID2^.....IDN FOR INPUT-MODE 4
ID1^ID2^.....IDN-H00H00 FOR INPUT-MODE 3
EACH IDENTIFIER IS SEPARATED BY AN '^' AND THE PLOT
DESCRIPTION SEPARATED FROM THE DATA BY A '-'. THE RECORD
LENGTH CANNOT EXCEED 16 CHARACTERS. THERE CAN BE A MAXIMUM OF 5
IDENTIFIERS, INCLUDING THE TEST NUMBER IF USED.

2) AT THE END OF DOWNLOADING A FILE OR TEST, A CONTROL REPORT IS
PRINTED GIVING:
2.1) NO. OF HEADER RECORDS DOWNLOADED
2.2) NO. OF PROMPTS DOWNLOADED
2.3) NO. OF RECORDS TRANSMITTED.
2.4) NO. OF RECORDS SUCCESSFULLY DOWNLOADED.
2.5) NO. OF RECORDS IN ERROR.
2.6) LIST OF RECORDS NOT SUCCESSFULLY DOWNLOADED.
2.7) LIST OF RECORDS WHICH WERE RETRANSMITTED.

TERMINATION:

TO TERMINATE THE PROGRAM ANSWER 'N' TO INPUT QUESTION 12.

FILES USED:

EITHER 1) FILE CREATED BY AN EDITOR.
2) UNFORMATTED FILE CREATED BY PROGRAM D6 (IDENTIFIER GENERATOR).
3) FILE CREATED BY A PROGRAM.
4) FILE CREATED BY THE DATAMYTE.

OPERATION:

1) DOWNLOAD HEADER RECORDS:

CONTROLS SHOULD BE SET AS FOLLOWS

CTRL 1 01 42 11
CTRL 2 15100200
CTRL 3 12412 2
CTRL 4 02002100

INPUT-MODE (C1-4) CAN BE SET TO 3 OR 4. CTRL 4 WILL VARY

ACCORDING TO NUMBER OF HEADERS REQUIRED - SEE DATAMYTE MANUAL FOR MORE DETAILS.

KEY CLEAR, CONTROL, RECEIVE.
DATAMYTE IS NOW READY TO ACCEPT HEADERS FROM D5 (DOWNLOAD) PROGRAM.

2) DOWNLOAD PROMPTS:

CONTROLS SHOULD BE SET AS FOLLOWS

CTRL 1	02 42 11
CTRL 2	15100200
CTRL 3	12412 2
CTRL 4	02002100

MEMORY STATUS MAY BE 0 OR 1 DEPENDING ON WHETHER HEADER RECORDS HAVE BEEN LOADED. INPUT-MODE (C1-4) CAN BE SET TO 3 OR 4. CTRL 4 WILL VARY ACCORDING TO THE NUMBER OF PROMPTS REQUIRED - SEE DATAMYTE MANUAL FOR MORE DETAILS.

KEY CLEAR, CONTROL, RECEIVE.
DATAMYTE IS NOW READY TO ACCEPT PROMPT STRINGS FROM THE D5 (DOWNLOAD) PROGRAM.

3) DOWNLOAD DATA:

IN ORDER TO DOWNLOAD THE DATAMYTE WITH A FILE OF IDENTIFIERS CONTROLS SHOULD BE SET AS ABOVE, EXCEPT C1-2 SHOULD BE 3 IF INPUT-MODE IS 4 AND SHOULD BE 4 IF INPUT-MODE IS 3.

KEY CLEAR, CONTROL, RECEIVE.
DATAMYTE IS NOW READY TO ACCEPT DOWNLOAD RECORDS FROM D5 (DOWNLOAD) PROGRAM.

4) TO COLLECT DATA:

IN ORDER TO RECORD DATA MANUALLY, C1-2 MUST BE SET TO 5.

5) TO TRANSMIT DATA TO A FILE ON THE PNFI PDP 11/44 USING PROGRAM D4 (UPLOAD).

CONTROLS SHOULD BE SET AS FOLLOWS.

CTRL 1	15 22 11
CTRL 2	15100000
CTRL 3	00006 0
CTRL 4	02002102

```
C1-4 WILL DIFFER ACCORDING TO INPUT-MODE USED.
CTRL 4 WILL DIFFER DEPENDING ON FORMATTING OPTIONS REQUIRED.

RUN [300,300]D4

KEY CLEAR, CONTROL, XMIT ON DATAMYTE.

AT END OF TRANSMISSION, TYPE CTRL+Z ON TERMINAL TO
TERMINATE FILE.
$
```

D5B(Download): This program is used to download information to the Datamyte memory which is stored on the portable Osborne computer floppy disks. It is written in BASIC language and is used for data collection at remote field locations.

See next page

```
D5B(DOWNLOAD)
200 PRINT"SET CONTROL 1 ON THE DATAMYTE TO 01 41 11"
210 PRINT"AND KEY CLEAR CONTROL RECEIVE"
220 INPUT "INPUT THE HEADER DATA";HEAD$
230 LPRINT HEAD$
240 PRINT
250 PRINT
260 PRINT"NOW SET CONTROL 1 ON THE DATAMYTE TO 12 41 11"
270 PRINT"AND KEY CLEAR CONTROL RECEIVE"
280 INPUT"NUMBER OF PROMPTS";NP
290 FOR I = 1 TO NP
300 INPUT"PROMPT NAME";PNAMES$
310 LPRINT PNAME$
320 NEXT I
330 PRINT
340 PRINT"INPUT THE NAME OF THE DISK"
350 PRINT"AND THE FILE WHERE THE DOWNLOAD"
360 PRINT"DATA HAS BEEN STORED"
370 PRINT"FOR E.G. B:E999A.DAT"
380 INPUT"NAME PLEASE";VNAME$
390 OPEN "I",#1,VNAME$
400 PRINT
410 PRINT
420 PRINT"NOW SET CONTROL 1 ON THE DATAMYTE TO 13 41 11"
430 PRINT"AND KEY CLEAR CONTROL RECEIVE"
440 PRINT
450 PRINT
460 PRINT"WHEN DATAMYTE IS READY,"
470 INPUT"PRESS (CR)";Z
480 FOR I = 1 TO 999
490 IF EOF(1) THEN 560
500 LINE INPUT#1,A$
510   PRINT A$
520 LPRINT A$
530 FOR J = 1 TO 50
540 NEXT J
550 NEXT I
560 CLOSE #1
570 END
```

Program D6(Identifier Generator):	This program creates an unformatted file of identifiers to be used to download the Datamyte. Each record to be downloaded can contain up to 5 identifiers.
Data Input:	All input is via the terminal. Questions are answered as requested.
Data Output:	See page 40
Program D6.FTN:	See next page

```
C   PROGRAM D6 (IDENTIFIER GENERATOR)
C
C   THIS PROGRAM CAN BE USED TO GENERATE SEQUENTIAL RECORDS OF IDENTIFIERS
C   FOR INPUT TO THE D5 (DOWNLOAD) PROGRAM.
C
C   THE PROGRAM WAS OBTAINED FROM AGRICULTURE CANADA
C   AND HAS BEEN  DOCUMENTED, MODIFIED, CHECKED AND MADE OPERATIONAL
C   ON THE PDP 11/44 AT P.N.F.I. BY WILLIAM KEAN IN OCTOBER 1982.
C
C   THE PROGRAM PROMPTS FOR:
C     1.  NAME OF OUTPUT FILE
C     2.  NUMBER OF IDENTIFIERS TO BE GENERATED IN EACH O/P RECORD (MAX 5)
C     3.  START NO, END NO AND INCREMENT TO BE USED TO GENERATE EACH IDENTIFIER
C   E.G. NUMBER OF IDENTIFIERS = 2
C        IDENTIFIER 1 STARTS AT 1, ENDS AT 5, INCREMENT 1
C        IDENTIFIER 2 STARTS AT 10, ENDS AT 12, INCREMENT 2
C   WOULD GENERATE A FILE AS FOLLOWS:
C     1   10
C     1   12
C     2   10
C     2   12
C     3   10
C     3   12
C     4   10
C     4   12
C     5   10
C     5   12
C   THIS PROGRAM GENERATES SEQUENTIAL RECORDS FOR INPUT
C   TO THE DATAMYTE D5 (DOWNLOAD) PROGRAM
C
      INTEGER*4 SNO(5),ENO(5),INC(5),PID(5)
      BYTE FNAM(20),YES,REPLY
      DATA YES/'Y'/
      WRITE(5,100)
  100 FORMAT (/1H$,15X,'NAME OF OUTPUT FILE:')
      READ(5,150)NCHARS,FNAM
  150 FORMAT(Q,20A1)
      FNAM(NCHARS+1)="0
      OPEN(UNIT=1,NAME=FNAM,STATUS='NEW',FORM='UNFORMATTED',
     -     RECORDTYPE='FIXED')
      WRITE(5,300)
  300 FORMAT(/1H$,'NO. OF INDENTIFIERS TO BE GENERATED MUST BE < 6 :')
      READ(5,310)N
  310 FORMAT(I2)
      DO 320 I=1,5
      SNO(I)=1
      INC(I)=1
  320 CONTINUE
      DO 400 I=1,N
      WRITE(5,330)I
  330 FORMAT(1H0,23X,'IDENTIFIER ',I2)
      WRITE(5,340)
  340 FORMAT(1H$,23X,'    START NO:')
      READ(5,345)SNO(I)
  345 FORMAT(I8)
      WRITE(5,350)
  350 FORMAT(1H$,23X,'      END NO:')
      READ(5,345)ENO(I)
      WRITE(5,360)
  360 FORMAT(1H$,23X,'   INCREMENT:')
      READ(5,345) INC(I)
  400 CONTINUE
```

```
      DO 800 II=SNO(1),ENO(1),INC(1)
      IF(N.LT.2) GO TO 500
      DO 750 JJ=SNO(2),ENO(2),INC(2)
      IF(N.LT.3) GO TO 500
      DO 700 KK=SNO(3),ENO(3),INC(3)
      IF(N.LT.4) GO TO 500
      DO 650 LL=SNO(4),ENO(4),INC(4)
      IF(N.LT.5) GO TO 500
      DO 600 MM=SNO(5),ENO(5),INC(5)
  500 PID(1)=II
      PID(2)=JJ
      PID(3)=KK
      PID(4)=LL
      PID(5)=MM
      IRCT=IRCT+1
      WRITE(1)PID
      IF(N.LT.2) GO TO 800
      IF(N.LT.3) GO TO 750
      IF(N.LT.4) GO TO 700
      IF(N.LT.5) GO TO 650
  600 CONTINUE
  650 CONTINUE
  700 CONTINUE
  750 CONTINUE
  800 CONTINUE
      WRITE(5,900)IRCT
  900 FORMAT(1H0,'NUMBER OF RECORDS GENERATED = ',I6)
      REWIND(UNIT=1)
      WRITE(5,910)
  910 FORMAT(/1X'DO YOU WISH TO HAVE THE FILE PRINTED AT YOUR
     + TERMINAL Y/N:    '$)
      READ (5,920) REPLY
  920 FORMAT(A1)
      IF (REPLY.NE.YES) GO TO 980
      WRITE(5,924)
  924 FORMAT(/1X'THE NUMBERS PRINTED REPRESENT
     + WHAT IS ACTUALLY IN THE FILE'
     +/1X'AND IN SEQUENCE, BUT ARE NOT IN THE FORMAT PRINTED,
     +/1X'AS THEY ARE UNFORMATTED IN THE FILE.'//)
  925 READ(1,END=980)PID
      WRITE(5,930)PID
  930 FORMAT(1X5I8)
      GO TO 925
  980 CLOSE(UNIT=1)
      STOP
      END
```

```
>RUN DL2:[300,300]D6

                NAME OF OUTPUT FILE:TEST.DAT

NO. OF INDENTIFIERS TO BE GENERATED MUST BE < 6 :2

                        IDENTIFIER  1
                            START NO:1
                              END NO:3
                           INCREMENT:1

                        IDENTIFIER  2
                            START NO:1
                              END NO:4
                           INCREMENT:1

NUMBER OF RECORDS GENERATED =     12

DO YOU WISH TO HAVE THE FILE PRINTED AT YOUR TERMINAL Y/N:   Y

THE NUMBERS PRINTED REPRESENT WHAT IS ACTUALLY IN THE FILE
AND IN SEQUENCE, BUT ARE NOT IN THE FORMAT PRINTED,
AS THEY ARE UNFORMATTED IN THE FILE.

       1       1       0       0       0
       1       2       0       0       0
       1       3       0       0       0
       1       4       0       0       0
       2       1       0       0       0
       2       2       0       0       0
       2       3       0       0       0
       2       4       0       0       0
       3       1       0       0       0
       3       2       0       0       0
       3       3       0       0       0
       3       4       0       0       0
TT11  --  STOP
```

APPENDIX II

Communication between resident minicomputer, portable microcomputer, and the Datamyte

This appendix includes detailed instructions of the following topics:

(1) Moving data from the PDP 11/44 to the Osborne.

(2) Controlled transmisson of data from the Osborne to the Datamyte.

(3) Uploading data from the Datamyte to the Osborne.

(4) Controlled transmission of data from the Osborne to the PDP 11/44.

Moving data from PDP 11/44 to Osborne

A file containing data to be downloaded can be moved from the resident computer to the Osborne by means of a program called Micro-Link. This program allows the user to establish a communications link between the resident and microcomputer. To invoke Micro-Link, place the appropriate diskette in disk drive A of the Osborne and press RETURN. The HELP logo and menu will appear. Press the ESC key to enter the CP/M operating system. When CP/M has loaded (A>), type ML followed by a RETURN to enter Micro-Link. The Micro-Link logo will appear. After the disk activity light on A (red light) has gone out, press any key to obtain the set-up display. Depending upon whether you are using a 1200 baud modem or a 300 baud acoustic coupler (telephone line), select the appropriate speed from the list displayed. A menu of 15 commands will appear along with the Micro-Link logo (:). The program is now ready to establish the link between the Osborne and the PDP 11/44. The following commands will allow you to move the data file from the PDP 11/44 to the Osborne.

```
:7   (no LF with CR)
:4   (clears copy buffer to 0's)
:35  (do not edit copy buffer)
:1   (establish link with host PDP 11/44)

>HEL
>PASSWORD
>Sign on-information
>PIP TI: = EXPERIMENT.DAT (At this point, press ESC on Osborne to
                          return to Micro-Link)

:3 (turns on the copy buffer)
:8 (turns off the screen formatting)
:1 (back to host PDP 11/44)

(press the return key to begin transmitting data to the Osborne)

When the transmission is finished, press the ESC
key to return to Micro-Link.
```

:3 (turns off the copy buffer)
:5.B:FILENAME.DAT (save the buffer under the file-name on disk B)
:1 (re-establish communication with host computer for wrap-up commands)

>BYE
(Press ESC key)

:10 (exit Micro-Link).

The data have now been transferred from the resident host (PDP 11/44) to a file on the portable Osborne. These data can now be accessed in the field and transferred to the Datamyte as required.

Controlled transmission of data from Osborne to Datamyte

To move data from the diskette to the Datamyte, it is necessary to invoke a program called D5B(Download), written in MBASIC. To load the program, place the appropriate disk in drive A and press return. The MBASIC logo will appear on the screen. Load the program from disk B into memory by issuing the following command:

LOAD "D5B

When the program has been sucessfully loaded, an ok will appear on the screen. At this point, take the diskette containing the program out of drive B and insert your data diskette. Attach the Datamyte to the RS232C port using the appropriate interface cables. Clear the Datamyte memory and set the various functions as requested by the program D5B. To get program D5B to execute, simply type RUN followed by a <CR>. After successful execution of program D5B, the data will have been downloaded from diskette to the memory of the Datamyte. The Datamyte is now ready to store data collected from the field experiment.

Uploading data from the Datamyte to the Osborne

To move data from the Datamyte to the Osborne it is necessary to invoke the program Micro-Link. To do this, insert the diskette containing the program into drive A. Press return and wait for the Help logo to appear. Press the ESC key to enter CP/M. When you have the CP/M prompt (A>) type ML followed by a return to enter the program Micro-Link. When the Micro-Link logo appears, press any key to obtain the set-up menu. Select the 1200 baud option by pressing A. The screen should change to display 15 menu-options along with the Micro-Link prompt (:). At this point, physically link the Datamyte to the RS232C port using the appropriate AMP cords and place the four control functions of the Datamyte to the following settings:

DATAMYTE

CONTROL	SETTING
1	15 22 11
2	0510000
3	00000 0
4	01001101

When this has been accomplished, execute the following set of instructions to the Micro-Link program to move the data from the Datamyte to the Osborne.

```
:2.2            (Turn Osborne into the host computer)
:20             (Send a line feed after a <CR>)
:4              (Clear the copy buffer to 0's)
:35             (Do not edit copy buffer)
:8              (Turn screen formatting off)
:3              (Toggle copy buffer switch to on)
:1              (Exit to peripheral [DM in this case])
```

Key, CLEAR, CONTROL, XMIT
ON DATAMYTE

When data have been transmitted, a bell will ring; press ESC to re-enter the program Micro-Link.

```
:3              (Toggle copy buffer switch to off)
```

NOTE: Ensure that a formatted diskette is in residence in drive B. Data from the Datamyte will be stored on this diskette.

```
:5.B:FILENAME.DAT (Save data on drive B under filename)
```

Data have now been safely stored on the diskette in drive B of the OS-1. To exit from the Micro-Link program, type a 10 followed by a <CR>.

Controlled transmission of data from Osborne to PDP 11/44

When the field work has been completed, it is necessary to move the data from the diskette of the Osborne to the resident computer for processing and permanent storage. To do this, we will invoke the program Micro-Link again. To load Micro-Link, place the appropriate diskette into drive A and press return. When the Micro-Link logo is displayed, press any key. There will be a menu of items from which the baud rate etc. can be modified. Choose the baud rate correct for the transmitting device you are using. (Usually, 1200 baud for direct-link modems and 300 baud for dial-up acoustic couplers). After selecting the baud rate a menu of 15 items should be displayed along with the Micro-link prompt (:). Enter the following series of commands to transfer the data from a diskette to the resident mainframe (PDP 11/44 in this case).

```
     :7                  (no LF with CR)
     :1                  (Establish link with host)

>HELLO (Account or Name)
>Password
>Sign on-information
>EDT FILENAME.DAT
 Input file does not exist
*I

     (Press ESC to return to Micro-Link)
     :6.B:FILENAME.DAT (Move file from disk to memory)
                       (Answer Y to question)
     :24.8             (8 clock units time delay)
     :21               (delay after CR)
     :42.XXXX          (where XXXX is location from 6)
     :47               (use address for <EOF>)
     :17               (accept P/R switch)
     :9                (send file)

                         DATA ARE BEING TRANSFERRED

     :1                (re-establish link with mainframe)

<cr>
*EXIT
Wrap-up Commands
LOGOFF INFORMATION

     Press the ESC key to re-enter Micro-Link
     :10 (Exit from Micro-Link)
```